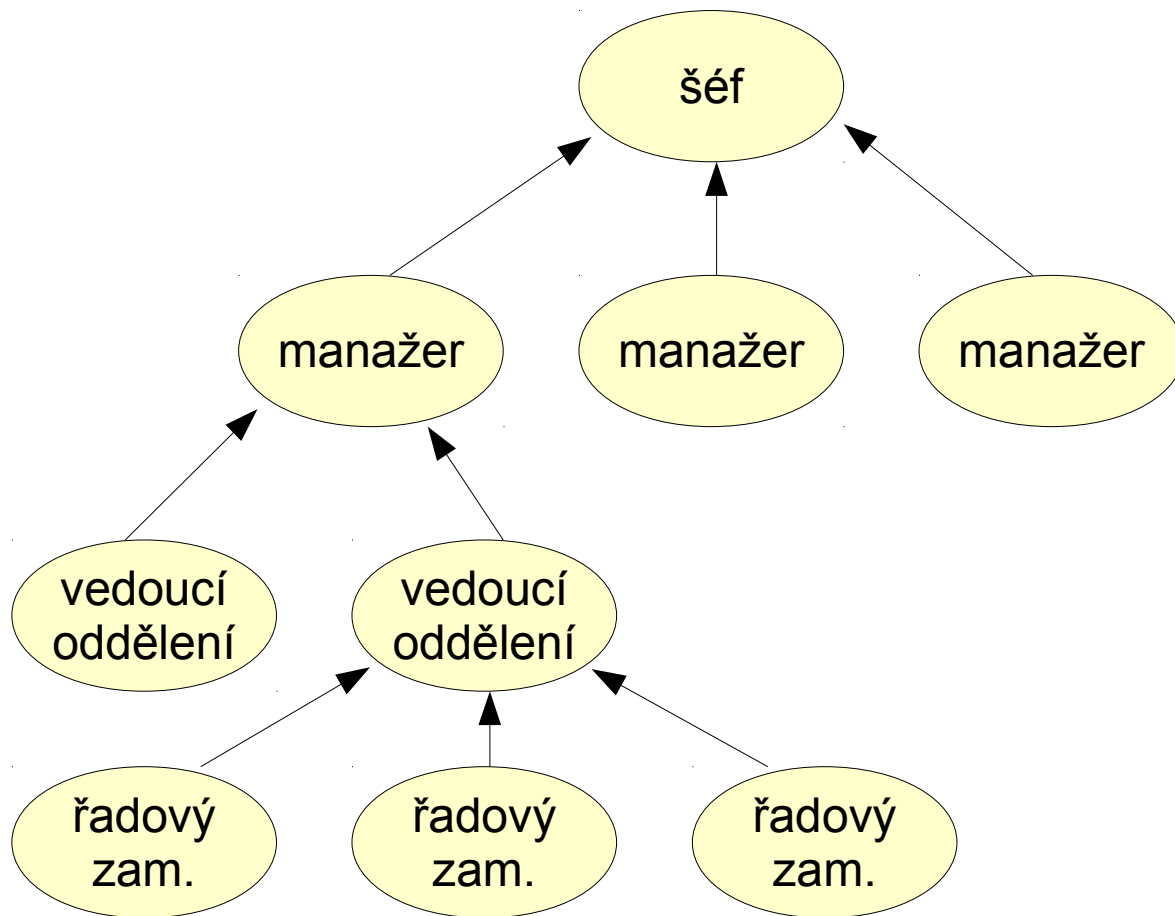


Hierarchické dotazy

Hierarchické dotazy

- Kde všude využijete hierarchické dotazy? Kdekoli máte stromovou strukturu v databázi v této podobě:



AERO.ZAMESTNANEC		
P	* ZAMESTNANEC_ID	NUMBER (7)
	* ZAMESTNANI_ID	NUMBER (3)
F	NADRIZENY	NUMBER (7)
	* JMENO	VARCHAR2 (32 CHAR)
	* PRIJMENI	VARCHAR2 (32 CHAR)
	PLAT	NUMBER (8)
	* DATUM_NASTUPU	DATE
	DATUM_UKONCENI	DATE
	* AKTIVNI	CHAR (1 BYTE)
PK_ZAMESTNANEC (ZAMESTNANEC_ID)		
◆ PK_ZAMESTNANEC (ZAMESTNANEC_ID)		
◆ IX_PRACUJE_JAKO (ZAMESTNANI_ID)		
◆ IX_MA_NADRIZENEHO (NADRIZENY)		
◆ INDEX_JMENO_ZAM (JMENO)		

Hierarchické dotazy

- Pokud tabulka obsahuje data se stromovým uspořádáním, lze vybírat záznamy v hierarchickém pořadí pomocí dotazu se syntaxí:

```
SELECT <seznam výstupních sloupců>  
FROM <seznam tabulek>  
[WHERE <podmínka pro omezení vybraných záznamů>]  
[START WITH <podmínka pro výběr kořenů>]  
CONNECT BY [NOCYCLE] <podmínka pro výběr  
potomků>  
[ORDER BY <seznam řazených sloupců>];
```

- Často se používá pro výpis:
 - organizační struktury společnosti,
 - přidělených práv uživatelů v hierarchické struktuře uživatelských rolí,
 - dokumentů v adresářové struktuře.

Hierarchické dotazy

- START WITH <podmínka> – udává kořen(y) hierarchického dotazu – záznam(y) v první úrovni, od kterého (kterých) včetně se začínají vypisovat podřízené záznamy.
- CONNECT BY <podmínky> – definuje provázání nadřízeného záznamu s podřízeným (rodič – potomek). Pro každý kořenový záznam se vyhledají přímí potomci, kteří vyhovují podmínce v klauzuli CONNECT BY, kde se záznam předka označuje klíčovým slovem PRIOR (PRIOR <výraz> = <výraz>, <výraz> = PRIOR <výraz>). Dále se postupuje rekurzivně (pro prohledání celé hierarchie).
 - NOCYCLE – dotaz bude vracet i záznamy, které mají potomky, které jsou zároveň předky (zacyklení). Pokud budou data obsahovat smyčky a NOCYCLE nebude uvedeno, dotaz skončí chybou.
- Na závěr jsou odstraněny všechny řádky, které nevyhovují podmínce v klauzuli WHERE, a poté je provedeno řazení podle ORDER BY.

Použití hierarchických dotazů

- Výpis organizační struktury od nadřízených:

```
SELECT z.zamestnanec_id, z.jmeno, z.prijmeni,  
        z.nadrizeny  
FROM zamestnanec z  
CONNECT BY PRIOR zamestnanec_id = nadrizeny;
```

- Výpis všech podřízených zaměstnanců s příjmením 'Herzog' (kořen hierarchického dotazu):

```
SELECT z.zamestnanec_id, z.jmeno, z.prijmeni,  
        z.nadrizeny  
FROM zamestnanec z  
START WITH prijmeni LIKE 'Herzog'  
CONNECT BY PRIOR zamestnanec_id = nadrizeny;
```

Použití hierarchických dotazů

- Výpis organizační struktury směrem od podřízených. Jako rodiče dotazu vezme zaměstnance na nejnižší pozici a postupně k němu vypíše jeho nadřízené:

```
SELECT z.zamestnanec_id, z.jmeno, z.prijmeni,  
        z.nadrizeny  
FROM zamestnanec z  
CONNECT BY zamestnanec_id = PRIOR nadrizeny;
```

- Výpis všech nadřízených zaměstnance s příjmením 'Kotrba' (kořen hierarchického dotazu):

```
SELECT z.zamestnanec_id, z.jmeno, z.prijmeni,  
        z.nadrizeny  
FROM zamestnanec z  
START WITH prijmeni = 'Kotrba'  
CONNECT BY zamestnanec_id = PRIOR nadrizeny;
```

Hierarchické dotazy - tipy

- V hierarchických dotazech lze použít pomocné pseudo-sloupce:
 - LEVEL – úroveň záznamu ve stromové struktuře.
 - CONNECT_BY_ISLEAF – udává, jestli je záznam listem (nemá žádné potomky).
 - CONNECT_BY_ISCYCLE – lze použít pouze v kombinaci s NOCYCLE. Vrací hodnotu 1, pokud má záznam potomka, který je zároveň předkem. Jinak vrací 0.
- Funkce CONNECT_BY_ROOT – vrací hodnoty z kořenového záznamu (nejvyššího předka) pro příslušný záznam.
- Funkce SYS_CONNECT_BY_PATH vrací cestu v hierarchii od kořene až k aktuálnímu záznamu.
- Klauzule ORDER SIBLINGS BY – řazení záznamů na stejné úrovni.

Hierarchické dotazy – příklad

```
SELECT z.zamestnanec_id, z.jmeno, z.prijmeni,  
       z.nadrizeny, zi.nazev_pozice,  
       LEVEL-1 AS pocet_nadrizenych,  
       CONNECT_BY_ISLEAF AS nema_podrizene,  
       CONNECT_BY_ISCYCLE AS je_smycka,  
       CONNECT_BY_ROOT (jmeno||' '||prijmeni) AS Sef,  
       SYS_CONNECT_BY_PATH(prijmeni, '/') AS cesta  
FROM zamestnani zi, zamestnanec z  
WHERE zi.zamestnani_id = z.zamestnani_id  
START WITH zi.nazev_pozice LIKE 'Ředitel aerolinek'  
CONNECT BY NOCYCLE PRIOR zamestnanec_id = nadrizeny  
ORDER SIBLINGS BY prijmeni, jmeno;
```